

EM574494457

EL764085351

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Handheld Computing Device With External
Notification System**

Inventor(s):

William Vong

Chad Schwitters

ATTORNEY'S DOCKET NO. MS1-155US

1 **TECHNICAL FIELD**

2 This invention relates to portable handheld computing devices, such as
3 handheld personal computers (H/PCs). More particularly, this invention relates to
4 an external notification system for handheld computing devices.

5 **BACKGROUND OF THE INVENTION**

6 Small, handheld computing devices have been steadily growing in
7 popularity in recent years. The devices go by different names, including palmtops,
8 pocket computers, personal digital assistants, personal organizers, and the like.
9 This disclosure is primarily directed to a class of computing devices referred to as
10 handheld personal computers, or "H/PCs", although aspects of this invention can
11 be implemented other types of handheld computing devices.

12 H/PCs are small, pocket-sized devices having an LCD (liquid crystal
13 display) with a touch-sensitive screen, a stylus to enter data through the screen,
14 and an input device such as a keypad or miniature QWERTY keyboard. H/PCs
15 have a microprocessor, memory, and are capable of running an operating system
16 and one or more applications on the operating system. Microsoft Corporation
17 recently released the Windows® CE operating system for use on H/PCs, which is
18 a scaled-down version of its popular Windows® operating systems manufactured
19 for personal computers.

20 One of the most desirable characteristics of H/PCs is their portability. The
21 compact, portable H/PCs provide a user with real computer-like applications—
22 such as email, PIM (personal information management), Internet browser,
23 spreadsheet, word processing. A traveling user can receive email messages,
24 schedule meetings or appointments, and browse the Internet from the H/PC.

1 Some handheld computing devices can notify a user of a scheduled event, if
2 they are turned on. The device plays an alarm sound, or pops-up a dialog box, to
3 alert the user of the event. However, many handheld computing devices have no
4 means of notifying a user when they are turned off, which is normally the case to
5 conserve power. While some handheld computing devices might be configured to
6 wake up and sound an alarm, such devices typically time out the alarm after a short
7 period. As a result, the user can miss the alarm because it terminates before being
8 noticed. In addition, audio alarms may, on occasions, be too faint for the
9 surrounding environment (e.g., an alarm might be overpowered by noise in an
10 airplane flight) or not sufficiently strong to command a user's attention when the
11 user is not immediately next to the device.

12 It would be advantageous to develop a notification system for handheld
13 computing devices, such as H/PCs, that notifies a user when an event occurs
14 regardless of whether the device is on or off, open or closed, pocketed, or docked,
15 and which remains active until the user acknowledges it. It would also be
16 advantageous to develop a notification system that provide a lasting external
17 notification to the user, rather than a short-run alarm or a pop-up box that is not
18 externally visible.

19

20 **SUMMARY OF THE INVENTION**

21 This invention concerns a portable handheld computing device having a
22 notification system that alerts a user of an event regardless of whether the device is
23 on or off, open or closed, pocketed, or docked. The notification system has a
24 notification mechanism that is activated upon occurrence of the event and remains
25 active until the user acknowledges the activated mechanism.

1 According to an aspect of this invention, the notification mechanism is a
2 light emitting diode (LED) that is (by user option) turned on by the notification
3 system when an event occurs. The LED remains activated until the user takes
4 action to handle the event.

5 According to another aspect of this invention, the LED is mounted
6 externally on the handheld computing device. More particularly, the handheld
7 device has a casing with a lid and a base. The LED is mounted on the lid's upper
8 surface and wraps around to one of the end surfaces of the lid. In this manner, the
9 LED is visible to the user when the lid is closed onto the base (i.e., the device is
10 off) or when the lid is open (i.e., the device is on).

11 According to another aspect of this invention, the notification mechanism
12 also has a deactivation button mounted externally of the handheld computing
13 device. The user depresses the deactivation button to deactivate the LED (as well
14 as any other external signals that may be used). In one implementation, the LED
15 and deactivation button are integrated as a single component mounted on the
16 device lid.

17 According to yet another aspect of this invention, a notification program
18 runs on the handheld computing device and is callable by an application to help
19 schedule events. The notification program sets timers with the system clock,
20 which is always on even when the handheld computer is turned off. When a timer
21 expires, the system clock sends an interrupt to the notification program to wake up
22 the notification program so that it can turn on the LED. The LED is coupled to
23 power so that it can remain on and the notification program can go back asleep.
24 The LED continues emitting light until the user notices and presses the
25 deactivation button.

1 According to another aspect, the notification program places a taskbar
2 annunciator in the taskbar of an operating graphical user interface window when
3 an event is realized. After depressing the deactivation button in recognition of the
4 LED, the user can actuate the taskbar annunciator with a stylus or other means and
5 jump directly to the source of the event. For instance, actuating the taskbar
6 annunciator might open a window that describes an appointment, which is the root
7 of the event.

8 According to another aspect, the notification program supports a graphical
9 user interface that enables a user to set notification options specifying how external
10 notification is to operate. For instance, the user might prefer a flashing light in
11 combination with an alarm. The user can set these options through the user
12 interface. The options are saved in a structure that is accessed when a user
13 notification is set.

14 According to still another aspect, the notification program is called by the
15 applications on the handheld computing device through an application program
16 interface (API). The API defines a time parameter that specifies when the user
17 notification should occur and a type parameter that references the structure
18 containing the user-defined notification options.

19

20 **BRIEF DESCRIPTION OF THE DRAWINGS**

21 The same reference numbers are used throughout the drawings to reference
22 like components and features.

23 Fig. 1 is a perspective view of a handheld computing device in an open
24 position.

1 Fig. 2 provides front, side, and end elevation views of the handheld
2 computing device in a closed position.

3 Fig. 3 is a block diagram of the handheld computing device.

4 Fig. 4 is a block diagram of the hardware/software architecture of a
5 notification system implemented in the handheld computing device.

6 Fig. 5 is a diagrammatic illustration of a graphical user interface window
7 embodied as an “options” dialog box.

8 Fig. 6 is a diagrammatic illustration of a screen image presented on a
9 display of the handheld computing device.

10 Fig. 7 is a diagrammatic illustration of a graphical user interface window
11 embodied as a “notification” dialog box.

12

13 **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT**

14 Figs. 1 and 2 show a handheld computing device 20. As used herein,
15 “handheld computing device” means a small general computing device having a
16 processing unit that is capable of running one or more application programs, a
17 display, and an input mechanism that is typically something other than a full-size
18 keyboard. The input mechanism might be a keypad, a touch-sensitive screen, a
19 track ball, a touch-sensitive pad, a miniaturized QWERTY keyboard, or the like.

20 The handheld computing device 20 of Figs. 1 and 2 is embodied as a
21 handheld personal computer (H/PC). The terms “handheld computing device”
22 and “handheld personal computer” (or H/PC) are used interchangeably throughout
23 this disclosure. However, in other implementations, the handheld computing
24 device may be implemented as a personal digital assistant (PDA), a personal
25 organizer, a palmtop computer, a computerized notepad, or the like.

1 Handheld computing device 20 has a casing 22 with a cover or lid 24 and a
2 base 26. The lid 24 is hingedly connected to the base 26 to pivot between an open
3 position (Fig. 1) and a closed position (Fig. 2). The handheld computing device 20
4 has an LCD (liquid crystal display) 28 with a touch-sensitive screen mounted in lid
5 24. The device is equipped with a stylus 30 to enter data through the touchscreen
6 display 28 and a miniature QWERTY keyboard 32, which are both mounted in
7 base 26. The handheld computing device 20 can also be implemented with a
8 wireless transceiver (not shown) such as an IR (infrared) transceiver and/or an RF
9 (radio frequency) transceiver. Although the illustrated implementation shows a
10 two-member H/PC 20 with a lid 24 and a base 26, other implementations of the
11 H/PC might comprise an integrated body without hinged components, as is the
12 case with computerized notepads (e.g., Newton® from Apple Computers).

13 In the above respects, the H/PC 20 is of conventional design and will not be
14 described in detail. Many manufacturers make suitable H/PCs. However, unlike
15 conventional H/PCs, the handheld computing device 20 of this invention is further
16 implemented with an external notification system.

17 In general, the external notification system is designed to alert a user of an
18 event regardless of whether the handheld computing device is presently on or off,
19 or whether the device is presently running a program. The notification system has
20 a notification mechanism that is activated upon occurrence of the event to alert the
21 user. The notification mechanism remains active until the user acknowledges it,
22 even if the handheld computing device is otherwise turned off. The notification
23 mechanism is an external, sensory perceptible mechanism that attracts the user's
24 attention when the device is on or off and regardless of whether the lid is open or
25 closed. The notification mechanism can be implemented in a variety of ways,

1 including a light, an audio generator, a vibration device, or other forms of sensory
2 perceptible mechanisms. In addition, these mechanisms may be used in
3 combination, or with other forms of sensory perceptible notices, such as a visual
4 dialog box on the display.

5 In the preferred implementation, the external notification mechanism
6 includes an externally mounted LED (light emitting diode) 40. When activated as
7 a result of an event, the LED is illuminated or made to flash. The LED 40 remains
8 activated until the user acknowledges it.

9 More particularly, the LED 40 is mounted on the external surface of the
10 H/PC 20 in a location that the user can view the light from different angles and
11 sides of the H/PC. In addition, the LED 40 is positioned to be seen when the lid 24
12 is open or closed. As shown in Fig. 2, the H/PC casing 22 has an upper surface 42
13 on lid 24, a lower surface 44 on base 26, a front side surface 46 (on both lid 24 and
14 base 26), an opposing back side surface 48 (on both lid 24 and base 26), and
15 opposing end surfaces 50 and 52 (on both lid 24 and base 26). The end surfaces 50
16 and 52 are dimensionally shorter than the elongated side surfaces 46 and 48.

17 The LED 40 is positioned on the upper surface 42 and wraps around an
18 upper corner to extend onto the end surface 50. The LED is raised on the end
19 surface 50 to be visible from the front. The LED may or may not be raised on the
20 upper surface 42. In this manner, the LED 40 can be viewed when the case 22 is
21 closed, either from above by viewing the LED portion on the upper surface 42 (for
22 instance when the H/PC is sitting on a desk), or from the side by viewing the LED
23 portion on the end surface 50 (for instance when the H/PC is slid upright into a
24 shirt pocket, purse, or briefcase). Additionally, the LED 40 can be viewed when
25 the case 22 is open (Fig. 1) by viewing the raised LED portion on the end surface

1 50. As an alternative to raising the LED on the end surface , the LED 40 may be
2 configured to wrap around to the inner surface of the lid 24 to be viewable when
3 the case 22 is open. . The LED itself might be configured in the illustrated shape,
4 or alternatively a normally shaped LED is configured to emit light into light-
5 transmissive tubing that conforms to the illustrated shape.

6 As shown in the side view of Fig. 2, the H/PC 20 also has a deactivation
7 mechanism to deactivate the LED 40 and any other external notification
8 mechanism. In the illustrated implementation, the deactivation mechanism is a
9 deactivation button 54 that is externally mounted on the end 50 to enable a user to
10 quickly locate and deactivate the external notification mechanisms, regardless of
11 whether the lid is open or closed. In a preferred embodiment, the deactivation
12 button 54 and LED 40 are integrated as a single component. The LED 40 can be
13 constructed to project slightly above the face of the deactivation button 54 to act as
14 a bumper to reduce the likelihood of accidental actuation. In other arrangements,
15 the button may be located separately from the LED.

16 Fig. 3 shows functional components of the handheld computing device 20.
17 It has a processor 60, a memory 62, a display 28, and a keyboard 32. The memory
18 62 generally includes both volatile memory (e.g., RAM) and non-volatile memory
19 (e.g., ROM, PCMCIA cards, etc.). An operating system 64 is resident in the
20 memory 62 and executes on the processor 60. The H/PC 20 preferably runs the
21 Windows® CE operating system from Microsoft Corporation. This operating
22 system is a derivative of Windows® brand operating systems, such as Windows®
23 95, that is especially designed for handheld computing devices. However, the
24 handheld computing device may be implemented with other operating systems.
25

1 One or more application programs 66 are loaded into memory 62 and run
2 on the operating system 64. Examples of applications include email programs,
3 scheduling programs, PIM (personal information management) programs, word
4 processing programs, spreadsheet programs, Internet browser programs, and so
5 forth. The H/PC 20 also has a notification manager 68 loaded in memory 62,
6 which executes on the processor 60. The notification manager 68 handles
7 notification requests from the applications 66, as is described below in more detail
8 with reference to Fig. 4.

9 The H/PC 20 has a power supply 70, which is implemented as one or more
10 batteries. The power supply 70 might further include an external power source
11 that overrides or recharges the built-in batteries, such as an AC adapter or a
12 powered docking cradle.

13 The H/PC 20 is also shown with three types of external notification
14 mechanisms: an LED 40, a vibration device 72, and an audio generator 74. These
15 devices are directly coupled to the power supply 70 so that when activated, they
16 remain on for a duration dictated by the notification mechanism even though the
17 H/PC processor and other components might shut down to conserve battery power.
18 The LED 40 preferably remains on indefinitely until the user takes action. The
19 current versions of the vibration device 72 and audio generator 74 use too much
20 power for today's H/PC batteries, and so they are configured to turn off when the
21 rest of the system does or at some finite duration after activation.

22 Fig. 4 shows the software and hardware architecture of a notification system
23 80 for the H/PC 20. The notification system 80 has a notification manager 68,
24 which is callable by the applications 66 through a user notification application

1 program interface (API). The API creates a new user notification or modifies an
2 existing one. It is given as:

3
4 PegSetUserNotification(hNotification, *AppName, *Time,
5 *UserNotification)
6

7 The API has four parameters, three of which are pointers. The
8 “hNotification” parameter specifies whether the call relates to creating a new user
9 notification or to modifying an existing notification. The parameter is either zero,
10 when a new notification is to be added, or contains an identity of the notification to
11 be modified.

12 The “AppName” pointer points to a null-terminated string that specifies the
13 name of the application 66 that owns the notification. The system uses the
14 application’s primary icon as the taskbar annihilator that is set by the notification
15 system to notify the user and enable immediate-click access to the application
16 responsible for the notification. The use of a taskbar annihilator is described
17 below with reference to Fig. 6. The “Time” pointer points to the system time
18 structure that specifies a time when the notification should occur. The
19 “UserNotification” pointer points to a Peg_User_Notification structure that
20 describes the events that are to occur when the notification time is reached.

21 More particularly, the Peg_User_Notification structure is a user
22 configurable structure that holds notifications options preferred by the user. The
23 application passes the user’s preferences to the system when scheduling an event
24 by specifying the address of the structure. Each application passes in a structure
25 that applies only to it, so notifications for different applications can be

1 differentiated. Similarly, an application can pass in different structures for
2 different events, so individual notifications can be differentiated.

3 The Peg_User_Notification structure contains information used to initialize
4 a dialog box user interface that is presented to the user when setting notification
5 options. Fig. 5 shows an example dialog box 100, which is supported by the
6 notification system 80. In this example, the dialog box 100 includes an option 102
7 as to whether to sound an alarm and a drop-down menu 104 that lists various
8 available alarm sounds, such as “Beep”. The drop-down menu 104 might contain
9 identities of other “.wav” files containing different alarm sounds the user might
10 prefer. A repeat option 106 is also provided so that the user can elect to have the
11 alarm repeated.

12 The dialog box 100 also has an option 108 that allows a user to enable or
13 disable a dialog box that can be displayed describing the notification when it goes
14 off. An option 110 allows the user to elect whether to have the LED 40 flash, or
15 not, during notification.

16 It is noted that the dialog box 100 is provided for example purposes, and
17 other options may be included. For instance, the dialog box 100 might include an
18 option to enable/disable the vibration device 72, or to combine the external
19 notification mechanisms so the LED 40, vibration device 72, and alarm 74 all go
20 off at different times in a continuous cycle.

21 Once the user fills in the dialog box 100, the options are stored in the
22 Peg_User_Notification structure. This structure is provided below:

```
1 UserNotificationType {
2     DWORD ActionFlags;
3     TCHAR *DialogTitle;
4     TCHAR *DialogText;
5     TCHAR *Sound
6     DWORD MaxSound;
7     DWORD Reserved;
8 }
```

9
10 The “ActionFlags” parameter specifies the actions to take when a
11 notification event occurs. This parameter is a combination of bit flags, as set forth
12 in the following table.
13

| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|-------------|---|----|----|----|----|----|----|----|----|----|----|
| Value | Meaning | | | | | | | | | | |
| PUN_LED | Flash LED. | | | | | | | | | | |
| PUN_VIBRATE | Vibrate the Device. | | | | | | | | | | |
| PUN_DIALOG | Display the user notification dialog box. When this structure is passed to the PegSetUserNotification API, the DialogTitle and DialogText pointers point to the title and text. | | | | | | | | | | |
| PUN_SOUND | Play the sound file identified by the Sound pointer. | | | | | | | | | | |
| PUN_REPEAT | Repeat sound file for T seconds. | | | | | | | | | | |

20 The “DialogTitle” pointer specifies the title of the user notification dialog
21 box. If this parameter is null, no dialog is displayed. The “DialogText” pointer
22 specifies the text of the user notification dialog box. If this parameter is null, no
23 dialog is displayed. The “Sound” pointer references a buffer that contains the
24 unqualified name of a sound file to play. The “MaxSound” parameter specifies the
25

1 maximum length of the string that can be copied into the sound buffer. The
2 “Reserved” parameter is reserved for future use and is presently set to zero.

3 With reference again to Fig. 4, the notification manager 68 passes a
4 command to an alarm manager 82 to set an alarm for a notification event. The
5 alarm manager 82 generates a set alarm command that is output to the real-time
6 clock 84 to tell the clock to set an alarm at the scheduled time of the notification
7 event. When the clock reaches the event time, it notifies an interrupt manager 86
8 through an interrupt. The interrupt manager 86 informs the notification manager
9 68 that the time of the event has arrived. The notification manager 68 then sends
10 out activation commands to an LED driver 88 to turn on LED 40. A button device
11 driver 90 is also provided to handle interrupts generated when the notification
12 button 54 is depressed to disable the LED 40.

13 To explain the architecture in the context of an example situation, suppose
14 the user starts a calendar application 66 and schedules an event notification for
15 8:00 AM. The user clicks on an “options” button to bring up the dialog box 100
16 (Fig. 5) to ensure that the LED and alarm are both enabled. The user then closes
17 the dialog box 100 and saves the clock settings. The calendar application 66 calls
18 the notification manager 68 using the PegSetUserNotification API, which includes
19 a pointer to the structure containing information specifying how the LED and
20 alarm are to behave.

21 The notification manager 68 stores the scheduled notification and examines
22 it in light of any other scheduled user notifications to determine which notification
23 is associated with the next chronological event to occur. Suppose that the
24 calendar notification is next to occur. The notification manager 68 then calls the

alarm manager 82, which in turn sets a hardware alarm for 8:00 AM in real-time clock 84. The user can then exit the application 66 and turn off the device.

At 8:00 AM, the real-time clock 84 sends an interrupt to interrupt manager 86. The interrupt manager 86 identifies the interrupt as clock-related, and routes the interrupt to the alarm manager 82. Upon receiving the interrupt, the alarm manager 82 pulses the event that the notification manager 68 has been waiting on. The notification manager 68 determines that the event is associated with the 8:00 AM calendar alarm and checks the user options to decide how the user wishes to be notified. Assuming that the user wants the light to flash and an alarm to sound, the notification manager 68 calls the LED device driver 88 to start flashing the LED 40 and concurrently plays the selected alarm. The notification manager 68 also creates a taskbar annihilator.

Suppose that the user is not present at 8:00 AM. The handheld computing device 20 times out and turns off. Due to the direct coupling to power, however, the LED 40 remains flashing. The flash rate is selected to minimize power usage, without compromising usability. As one example, the LED flashes once every two seconds for $1/16^{\text{th}}$ of a second. The alarm preferably times out with the computing device 20 to conserve power, but it can be configured to repeat until the user acknowledges the notice.

When the user returns, he/she sees the flashing LED 40 and presses the deactivation button 54. Depressing button 54 generates an interrupt that is routed by interrupt manager 86 to keyboard device driver 90. The keyboard driver 90 instructs the LED driver 88 to turn off the LED 40. It is noted that the clock 84 and deactivation button 54 work the same if the H/PC 20 is already on. They still generate interrupts, only the interrupts do not wake up the device.

1 The deactivation button 54 is preferably implemented to turn off only the
2 external notification mechanisms (i.e., LED and alarm) because these mechanisms
3 can be annoying and consume a significant quantity of power. The taskbar
4 annunciator and optional dialog box remain intact and are not disabled by the
5 deactivation button 54. As a result, the user can open the H/PC and glean more
6 information concerning the notice from the taskbar annunciator and dialog box.

7 When the user opens the H/PC, the notification system 80 optionally causes
8 a taskbar annunciator for the calendar application to be displayed. Fig. 6 shows an
9 example screen image 120 showing the task bar 122 with a "Start" softkey 124 and
10 a time/date area 126. A taskbar annunciator 128 for the calendar application is
11 displayed in the time/date area 126. The annunciator 128 is the icon for the
12 calendar application, thereby immediately informing the user that the source of the
13 notification is the calendar application. The user can then click on the taskbar
14 annunciator 128 using the stylus to start the calendar application, as represented by
15 the calendar softkey 130.

16 The notification system 80 can optionally display a dialog box that explains
17 the notification to the user; this is faster than starting the originating application,
18 and provides a more noticeable notification if the H/PC is being used when the
19 notification event occurs. Fig. 7 shows an example dialog box 140, which contains
20 a message that informs the user of the 8:00 AM alarm. The user is also presented
21 with the option of accepting the alarm or rescheduling it for an additional five-
22 minute period. It is noted that tapping the taskbar annunciator 128 or a softkey in
23 the dialog box 140 deactivates the LED 40 and alarm, in the same manner as
24 pressing the deactivation button 54. Touching any annunciator or acknowledging
25

any dialog box turns off the external signals, since this tells the notification system that the user is aware of the notifications.

The notification system 80 can handle an arbitrary number of notifications from multiple applications. The notification manager 68 handles the scheduled events in temporal order. In some situations, the flashing LED might represent multiple notifications. Since a single flag bit controls the LED, the user presses the deactivation button 54 once to turn off the LED 40. The user can then open the device and examine the various annunciators to learn which applications are responsible for the notifications.

A full set of user notification APIs is attached to this disclosure as Appendix A.

Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as preferred forms of implementing the claimed invention.

17
18
19
20
21
22
23
24
25